

**GA-A22713**

**SOFTWARE DEVELOPMENT ON THE DIII-D  
CONTROL AND DATA ACQUISITION  
COMPUTERS**

**by  
B.G. PENAFLOR, B.B. McHARG, JR., and D. PIGLOWSKI**

**NOVEMBER 1997**

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, produce, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

GA-A22713

# SOFTWARE DEVELOPMENT ON THE DIII-D CONTROL AND DATA ACQUISITION COMPUTERS

by  
B.G. PENAFLORE, B.B. McHARG, JR., and D. PIGLOWSKI

This is a preprint of a paper to be presented at the 17th IEEE/NPSS Symposium on Fusion Engineering, October 6-11, 1997, San Diego, California and to be published in the *Proceedings*.

Work supported by  
the U.S. Department of Energy  
under Contract No. DE-AC03-89ER51114

GA PROJECT 3466  
NOVEMBER 1997

# Software Development on the DIII-D Control and Data Acquisition Computers\*

B.G. Penaflor, B.B. McHarg, Jr., and D. Piglowski

General Atomics

P.O. Box 85608, San Diego, California 92186-9784

**Abstract** — The various software systems developed for the DIII-D tokamak have played a highly visible and important role in tokamak operations and fusion research. Because of the heavy reliance on in-house developed software encompassing all aspects of operating the tokamak, much attention has been given to the careful design, development and maintenance of these software systems. Software systems responsible for tokamak control and monitoring, neutral beam injection, and data acquisition demand the highest level of reliability during plasma operations.

These systems made up of hundreds of programs totaling thousands of lines of code have presented a wide variety of software design and development issues ranging from low level hardware communications, database management, and distributed process control, to man machine interfaces.

The focus of this paper will be to describe how software is developed and managed for the DIII-D control and data acquisition computers. It will include an overview and status of software systems implemented for tokamak control, neutral beam control, and data acquisition. The issues and challenges faced developing and managing the large amounts of software in support of the dynamic and everchanging needs of the DIII-D experimental program will be addressed.

## 1. INTRODUCTION

The importance of home grown customized software systems in fusion research, particularly for the DIII-D program [1] at General Atomics (GA) cannot be overly stated. Because of the unique and dynamic nature of this type of research many of the systems which are responsible for directly controlling and monitoring the tokamak during operations and acquiring data during a plasma discharge have been custom developed "in-house" by GA programmers. Without these software systems, fusion research at DIII-D would be very difficult if not impossible to carry out. Much time and effort has been invested in the careful design, development and maintenance of these custom software systems in order to provide physicists and engineers at DIII-D with the most reliable and flexible systems possible to meet their research needs.

## 2. COMPUTER SYSTEMS OVERVIEW

A network of five Modcomp AEG computers (Fig. 1) each

running a real time variant of the UNIX operating system known as REALIX are used to perform most of the essential functions associated with the operation and acquisition of data for the DIII-D tokamak. The Modcomp AEG Computer System is a recent upgrade of hardware and software which was completed this year.

A Modcomp AEG "tokamak control" computer is dedicated to the task of controlling and monitoring various tokamak hardware systems including power, pressure and interlocks over a single CAMAC highway. A second AEG "data acquisition" computer [2] with two CAMAC highway connections is responsible for coordination with the DIII-D discharge cycle to set up and start the clocks and digitizers for acquiring data during a discharge. The data acquisition computer also reads up to 1600 signals at the end of a discharge to write the data to disk and serve the information to other computers for immediate analysis. Two AEG "Neutral Beam" [3] computers are used to control and acquire data for eight neutral beam sources over four CAMAC highway connections. A fifth AEG computer currently serves as a backup system and is also the host on which all of the software developed to run on these computers permanently resides. Each of the Modcomp AEG computers is a Motorola 88110 VME formatted cpu with memory ranging from 64 to 128 megabytes and it's own individual set of hard disk drives providing from 2 to 4 gigabytes of disk space per computer. Communication between computers is achieved over a twisted pair ethernet connection with a 10 megabit throughput.

The software for these computers consists of both vended applications and custom in-house developed codes. The AccessWare application is the primary vendor supplied software package used for CAMAC control and monitoring, and for the development of graphical interfaces for use by neutral beam and tokamak operators. The AccessWare application which runs under the REALIX/X11 operating system environment provides tools for quickly creating graphical user interfaces which can be easily tied to hardware modules connected over CAMAC. AccessWare is used on the AEG tokamak control and neutral beam computers. The Infinity relational database is another vended application provided by Logical Data that is used to manage hardware and discharge configuration data for all of the AEG computers.

---

\*Work supported by U.S. Department of Energy under Contract No. DE-AC03-89ER51114.

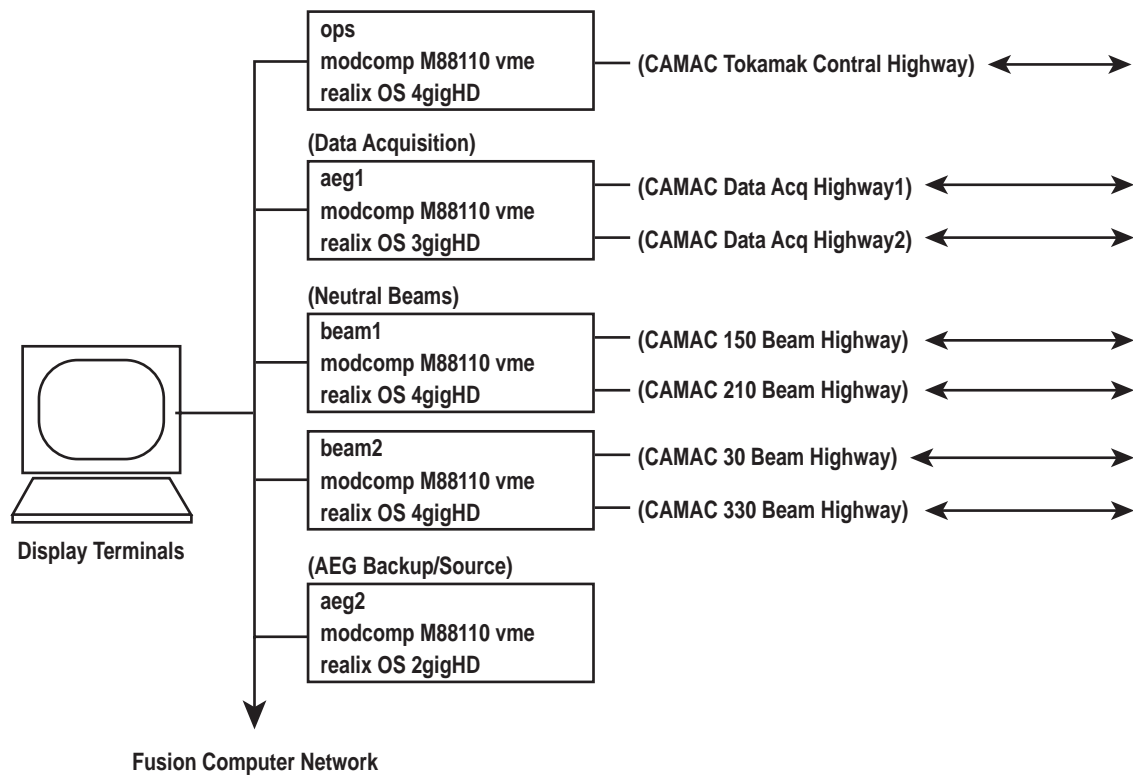


Fig. 1. AEG control and data acquisition computers.

The in-house developed software consists primarily of C programs and UNIX shell scripts. Most of these in-house developed applications require functionality specific to DIII-D which cannot be provided by “off the shelf” software products. A prime example of this is the Data Acquisition Software system developed in the C programming language by GA programmers. Because of the specific nature of the hardware and the setup of clocks and digitizers which depend on the ever changing experimental needs of the DIII-D program, the entire Data Acquisition system was custom developed to meet the specific needs of DIII-D. Other examples of custom codes developed for DIII-D include routines responsible for coordination of discharge cycle tasks, a software driver interface between the AccessWare application and the CAMAC hardware, numerous procedures which get called from AccessWare to perform tasks necessary for setting up, running and finishing up a discharge, a program to perform Residual Gas Analysis of the DIII-D vessel, an application for loading beam modulation and digitizer clock parameters, codes for analyzing various pieces of data collected in real-time such as vessel thermocouple temperatures, and numerous other codes.

### 3. SOFTWARE DEVELOPMENT ENVIRONMENT

The software developed for the AEG computers is maintained and managed by programmers in the General Atomics fusion computer group. It is the computer group's responsibility to organize, develop and make decisions on

all software running on the AEG computer systems. Though actual software development may also be performed by physicists and engineers outside of the computer group, all software written for these systems must be approved and coordinated by computer personnel in order to assure a system which is well organized and easily maintained. In addition to development and management of the custom software systems, the fusion computer group also performs all of the necessary system administration tasks for the AEG computers and is responsible for keeping the systems running at all times during plasma operations. The computer staff provides the experience and expertise to complement and assist physicists and engineers to get the best and most reliable use of the computer systems at DIII-D.

Software developers work closely with physicists and engineers to determine the programming needs of DIII-D. Requests for software work are submitted to the computer group from DIII-D personnel on a daily basis with tasks having the greatest impact on operations taking first priority. Many requests involve investigating and trouble shooting problems with existing software and can often be immediately resolved. Requests for software work which are more long term in nature are usually discussed in meetings involving both programmers and users.

Development of new software involves careful analysis of the software requirements and reviews of proposed designs for new software systems before coding begins.

Meetings incorporating “rapid prototypes” are held to gather and discuss software requirements and review software design plans. This advanced planning of software often helps to speed up overall development by making coding more focused and reducing time wasted in rewriting software modules or writing unnecessary codes. Code development may occur on any of the AEG computers, though actual programming during operations is somewhat restricted in order to prevent problems with the computer systems during this time. A backup AEG computer provides a platform upon which new software can be developed and tested even during operations with no adverse affects.

#### 4. SOFTWARE MANAGEMENT

At DIII–D much software programming effort goes into the maintenance and enhancement of existing codes. Maintenance usually comes in the form of dealing with unexpected results from an existing code usually arising from different unforeseen uses of the code or because of a change in the computer system affecting the codes function. Enhancements to existing codes range anywhere from simple one line changes in a single source file, to rewrites and creation of entire new sections of code to provide brand new capabilities. Software maintenance and enhanceability is a very important concern at DIII–D. As requests for different and new capabilities are made each day, the ability to integrate various changes into existing codes and still make the software understandable and easily maintained can become increasingly difficult. In some cases, an existing code can grow to become so large and complicated that it is sometimes simpler to rewrite the entire code from scratch than to attempt changes to the existing software.

A number of steps have been taken at DIII–D on the AEG computers to make existing codes more receptive to change. First, all source code for the AEG computers is maintained in a single area on one computer identified as the master source server using the standard UNIX sccs (source code control system) to manage changes to files. As straight forward and logical as this may seem, it is not uncommon to see source developed in which multiple copies with no control reside on various computer systems and media leading to countless problems in updating wrong versions of code and not knowing what changes have been made and tested in a code.

Another step made in making software more manageable for the AEG computers has been in maintaining consistency among applications. All programs developed in C have a Makefile and man page which are similar to one another. The Makefile can be used to easily build a program from the source files and also to install the program into a common application release area. The build and release process is further automated by a general

Makefile which can be used to build and install all of the applications for a specific AEG computer. To insure that problems in a recent change to a code can be immediately undone, a backup of the last working version of the code is automatically created each time a new release is made.

Maintaining consistency among applications is also accomplished through a careful choice of languages. Most of the applications developed for the AEG computers have been written in standard C or an agreed upon script language such as the Korn or C shells. Tcl/Tk and AccessWare have been adopted as standards for user interface development on all the AEG computers. While a free choice of languages and third party software packages might offer flexibility and immediate solutions to problems, this type of approach can also be very expensive to maintain in the long run. For example an application developed in the latest “fad” language or library downloaded from the internet might provide a quick solution to a software problem today, but as computers, operating systems and other software are upgraded, support for these languages or libraries could be difficult to impossible to obtain, making applications developed in them unmaintainable. A careful choice of languages and dependence on third party software can help lengthen the useful life of an application.

The creation of software libraries is another obvious and simple step taken to improve software maintainability. Separating out common routines has helped to reduce the sizes of codes, and well defined libraries have made it easier to develop new functionality. The libraries developed for the AEG computers includes a library of CAMAC functions for interacting with various types of CAMAC hardware, a library for interfacing GA custom codes with the AccessWare application, a library of routines to interface with the DIII–D pointname database and data archiving system, and a library of general useful C functions.

#### 5. RECENT DEVELOPMENTS

Most recent coding efforts on the AEG computer systems have concentrated on completion of the neutral beam systems upgrade. Programs originally coded in Fortran, making use of mainframe assembly language calls, were completely redesigned and rewritten in the C language to run on the new Modcomp AEG platform. Examples of this include the new neutral beam timing application, a plotting package for viewing beam waveforms digitized during a discharge, and a number of procedures for getting the neutral beams ready and firing during a discharge. In addition to this, the existing data acquisition software was modified to run on the beam computers.

A number of problems have been observed with the AccessWare application which have resulted in delays in

completion of the neutral beam upgrade and have also affected overall operations. Stopgap programming measures by GA programmers have helped to stabilize the computer systems although the dependence on AccessWare is still an important question being addressed.

On the tokamak control computer an application incorporating a Tcl/Tk user interface and a generic X11 plotting package was developed to collect and analyze thermocouple data taken during discharges and vessel bakes.

The amounts of data being collected by the data acquisition computer continues to increase with each operations period resulting in new records for total amounts of data collected per shot. A typical DIII-D shot now produces up to 200 megabytes of uncompressed data. The availability time to users of the freshly acquired data now ranges from 20 s to 6 min. Increasing demand for accessing data immediately after a shot has resulted in hardware upgrades to further increase the performance capabilities of the data acquisition system, including the addition of a second camac highway and the doubling of the AEG system memory.

## 6. CHALLENGES

With the rapid advances in computing hardware one of the first challenges in development of custom software systems is in keeping these systems up to date and in pace with the latest technology. As software systems and experimental needs grow, the need for faster more powerful processing capability is always expected. Computer system upgrades have become frequent and common place events at DIII-D in spite of the increasing level of efforts needed to do so. Replacing a computer system has rarely ever been as simple as plugging in a new computer and loading in and compiling the old applications on it. Many factors such as hardware specific aspects of old source codes, device driver compatibility, changes in the operating system and newer versions of vended third party applications often contribute to making computer hardware upgrades difficult and very lengthy experiences. Also the need to have an old system running while it is being upgraded adds to the difficulty of keeping

software in pace with hardware since it is most often the case changes are needed on the system being used. The problem can be likened to having to perform maintenance on your only car while you are driving it.

Another challenge to developing custom software systems is in reducing the reliance on individuals in order to keep the systems running. In many cases a software application is identified with a specific person or small group of people who are the only ones knowledgeable enough about it to keep it running. Standards for coding and documentation and exposure of different programmers to different tasks can help reduce over reliance on a few key individuals and increase the longevity of custom developed applications. Also good communication between those responsible for software development can contribute to software which can be maintained by a larger group of people.

## 7. SUMMARY

As the reliance on custom in-house developed applications in fusion research grows, so must the ability to effectively maintain and manage these codes. Quick solutions to software problems such as using the latest "fad" software which may provide immediate help now but could lead to expensive future maintenance problems need to be avoided. Writing volumes of code without careful planning and a well laid out software architecture is a quick recipe for a computer disaster or serious financial drain. Serious thought needs to be given to the overall software design and development process. An engineering discipline to writing code especially the large and complicated ones required in fusion research must be maintained and enforced in order to insure physicists and engineers of having the best and most reliable computer systems from which to conduct research.

## REFERENCES

- [1] J.L. Luxon, *et al.*, Plasma Phys. and Contr. Fusion **32**, (1990) 869.
- [2] B. McHarg, "Conversion of the central DIII-D data acquisition system to a UNIX based platform," unpublished, 1995.
- [3] D. Piglowski, *et al.*, "Use of the accessware interface/database software in the neutral beam control systems used by the DIII-D tokamak," these proceedings.